A Multi Agent System architecture to implement Collaborative Learning for social industrial assets ⋆

Kshitij Bakliwal * **Maharshi Harshadbhai Dhada** *
Adrià Salvador Palau ** **Ajith Kumar Parlikad** **
Bhupesh Kumar Lad *

* *Indian Institute of Technology (IIT) Indore, Khandwa Road, Simrol, Indore 453552 India*
** *DIAL, Institute for Manufacturing, 17 Charles Babbage Road, University of Cambridge, Cambridge CB3 0FS, UK*

Abstract:
The 'Industrial Internet of Things' aims to connect industrial assets with one another and subsequently benefit from the data that is generated, and shared, among these assets. In recent years, the extensive instrumentation of machines and the advancements in Information Communication Technologies are re-shaping the role of assets in our industrial systems. An emerging paradigm here is the concept of 'social assets': assets that collaborate with each other in order to improve system performance. Cyber-Physical Systems (CPS) are formed by embedding the assets with computing capabilities and linking them with their cyber models. These are known as the 'Digital Twins' of the assets, and form the backbone of social assets. Collaboration among assets, by allowing them to share and analyse data from other assets can make embedded computing algorithms more accurate, robust and reliable. This paper proposes a Multi Agent System (MAS) architecture for collaborative learning, and presents the findings of an implementation of this architecture for a prognostics problem. Collaboration among assets is performed by calculating inter-asset similarity during operating condition to identify 'friends' and sharing operational data within these clusters of friends. The architecture described in this paper also presents a generic model for the Digital Twins of assets. Prognostics is demonstrated for the C-MAPSS turbofan engine degradation simulated data-set (Saxena and Goebel (2008)).

Keywords: Cyber-Physical Systems, Industrial Internet of Things, Digital Twins, Collaborative Learning, Industry Automation, Multi Agent Systems, Distributed Computing.

1. INTRODUCTION

Accuracy, precision and cost-effectiveness of sensing technologies have improved in recent years, leading to extensive instrumentation of industrial assets and emergence of Big Data. Computers have become smart, compact, powerful and capable of operating over cloud servers. With advancement in communication technologies, low-cost transfer of significant amount of data over the internet is now possible (Ganchev et al. (2016)). Integrating all these developments with the objects of everyday use has led to a network of connected objects called the 'Internet of Things' (IoT). (Li et al. (2015))

Thanks to these technological advancements, up to date industrial assets are able to generate extensive data that reflects the system performance. Also, technologies such as RFID, smart cards, embedded systems, Wi-Fi, and Bluetooth communication have enabled automatised machine to machine communications to take place (Unland (2015)). Such technologies make it possible to harness the benefits of the data generated in industrial environments. This extension of the notion of IoT is termed as the 'Industrial Internet of Things' (IIoT) (Evans and Annunziata (2012); Xu et al. (2014)).

In the IIoT, each asset has a Digital Twin, which is its cyber model/ replica, containing the asset data acquired from various sources. This Digital Twin is part of a network of several other twins with their corresponding assets, together forming a network of industrial assets. The streams of data flowing into the twins are analysed by human experts or computational algorithms to get a perception about the surroundings, performance, and the health conditions of the assets (GE Digital (2017)).

⋆ This research work was sponsored by the project 'Building capacity in Collaborative Research for Advanced Manufacturing', funded by the Royal Academy of Engineering under the Newton Bhabha scheme. This research was supported by SustainOwner (Sustainable Design and Management of Industrial Assets through Total Value and Cost of Ownership), a project sponsored by the EU Framework Programme Horizon 2020, MSCA- RISE-2014: Marie Skodowska-Curie Research and Innovation Staff Exchange (Rise) (grant agreement number 645733 Sustain-owner H2020-MSCA-RISE-2014).
⋆⋆Both first authors Maharshi Harshadbhai Dhada and Kshitij Bakliwal have equal contributions in this paper.
⋆⋆⋆Corresponding author: Adrià Salvador Palau; e-mail id: as2636@cam.ac.uk

Embedded computers monitor and control the physical processes, usually with feedback loops. This way, physical processes shape the computations and vice-versa (Lee (2008)). The integration of computation with physical processes described above forms a Cyber-Physical System (CPS) (Jeschke et al. (2017)).

Further benefits can be harnessed by integrating human-like social networking capabilities into IoT. This notion of SIoT (Social Internet of Things), has been gaining momentum in the areas of product life-cycle management, traffic routing, and workplace help and support. The integration of social networks and IoT can be extended to improve system-level performance in an asset fleet. In this paper, we propose a quantitative method to identify groups of similar assets, or friends. Subsequently, condition data, and the diagnostics or the prognostics knowledge shared among these assets can improve the accuracy of the asset's computations (Li et al. (2018)). Several projects like the Toyota Friend, Nike+, Xlively, Social Web of Things, Evrythng, etc. have aimed at integrating IoT with a social-networking framework (Atzori et al. (2014)).

This paper describes a Multi Agent System (MAS) architecture for collaborative learning among social industrial assets. An implementation is shown for a prognostics problem using Saxena and Goebel (2008)'s turbofan engines degradation data-set (C-MAPSS). The presented architecture comprises three layers: Virtual Assets, Digital Twins, and a Social Platform. In the Virtual Asset layer, the data originating from different assets is standardised for its further analysis by a generic Digital Twin. Digital Twins are asset (type)-independent software components capable of adapting to a variety of assets. The proposed Digital Twin, once assigned to an asset generates asset-specific models using data from other assets in the fleet and the asset itself. The third layer comprises a Social Platform, with a role limited to enterprise level analysis, such as clustering similar assets or analysing machine data to generate fleet analytics. To enable collaboration between similar assets, the distances (similarities) among assets in the industrial system are evaluated and the data originating from similar assets is prioritised.

Section 2 reviews the literature. MAS and HMS approaches are covered in section 2.1, which also discusses existing architectures and frameworks based on collaborative learning or distributed decision-making. The benefits of collaborative learning and the SIoT paradigm are discussed in the subsection 2.2. The proposed architecture is explained in detail in section 3. An illustrative example of this architecture makes up the final section 4. The paper ends with conclusions (section 5).

2. LITERATURE REVIEW

2.1 Multi Agent Systems and Holonic Manufacturing Systems

Many paradigms have emerged to satisfy the requirements, and the challenges, of "new manufacturing" practices. Among which, agent-based manufacturing systems (Multi Agent Systems (MAS)), and Holonic Manufacturing Systems (HMSs) have received a lot of attention in academia and industry. MASs enable social behaviour of intelligent entities, through the capabilities of the agents forming the System. They are a broad software approach, unlike the manufacturing-specific approach of HMSs, focused on distributed control (Giret and Botti (2004)).

Decentralised architectures allow complex tasks to be divided into sub-tasks, allotted to the best suited agents. Decentralisation presents advantages like system robustness and agility, and elimination of data transfer lags. Enabled by MAS, these architectures have been used to tackle industrial problems. For example, Giordani et al. (2013) make use of a MAS approach to tackle the problem of production planning and scheduling. A two-layer hierarchical approach is employed by Mönch and Drießel (2005) to decompose the scheduling problem into simpler sub-problems. Christensen (2003) proposed an architecture where agents focus on deliberative tasks on a higher level, while lower-level agents focus on real-time constrained control tasks.

Bagheri et al. (2015) presented a step-wise approach to design a CPS architecture for an Industry 4.0 environment, and an adaptive clustering method for self-aware machines. Bagheri et al. (2015) discussed how one should progress from the smart-connection level to the configuration level while designing an architecture. Drawing from Bagheri's ideas, in this paper we present an architecture which can be implemented for any industrial system inspired by the Social Internet of Things paradigm. We also introduce the concept of 'Virtual Assets', an additional agent layer aimed to standardise the data flowing into the asset's Digital Twins. This standardisation of data makes it possible for us to have a generic model of Digital Twins, thus eradicating the need to tailor different Twins for every other asset in the industrial system.

2.2 Learning in Multi Agent Systems and the Social Internet of Things

Agents in an MAS need to keep learning in order to adapt to a dynamic environment. It has been shown that multi-agent learning can be reduced to single agent learning by considering the other agents in the system as part of the agent's environment. However, this may not always lead to an optimal solution and coordinated machine multi-agent learning becomes more important (Alonso et al. (2001)). An efficient way to achieve collaboration is to integrate social networking concepts into the Internet of Things. The assets in such a "Social Internet of Things" behave like social entities, sharing data and collaborating with one another to generate an optimal enterprise level solution. This paradigm, achieved by developing trust among assets which are "friends" with one another, permits navigability even after the point when number of nodes increase above those encountered in the traditional internet (Atzori et al. (2014)). Collaboration among assets in a system not only increases the responsiveness of the system, but also allows unseen events of importance to be broadcasted within a group of friends. This improves the accuracy of underlying algorithms by making a richer data-set available for training and prediction purposes. Ning and Wang (2011) describe an analogy of such systems with the social organisation of humans as: "each 'unit' human has its nervous systems made up of the same physical

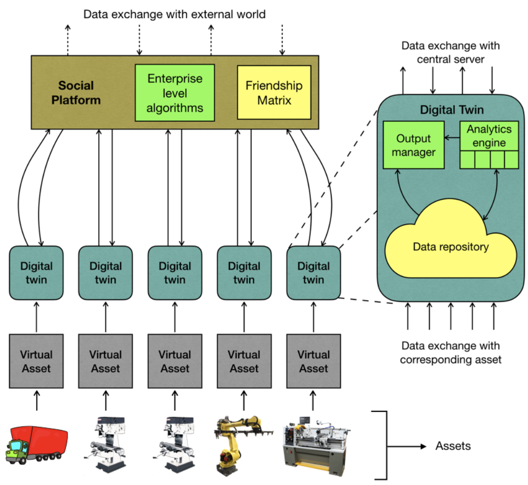

Fig. 1. Schematic layout of the MAS architecture.

components and operating laws, but individuals possess their own sophisticated and unique consciousness and behaviour".

Multi-agent collaborative learning can be implemented through several kinds of algorithms: social algorithms, swarm intelligence, etc. An example of social algorithms is evolutionary computation, a kind of Stochastic Search method among reward-based learning techniques (Alonso et al. (2001)). Other approaches that incorporate collaboration between agents are swarm intelligence techniques, that try to emulate the efficiency of foraging seen in natural systems such as those of bees or ant colonies. A relevant part of current research focuses on harnessing the technology capabilities by merging collaborative learning and Multi Agent Systems for the IIoT. However promising this idea may seem, there is a lack of an industrial-system architecture capable of integrating the social networking concepts with the IIoT. Our architecture addresses this gap by providing clearly organised levels, each suited for different analytics algorithms. Ours is a MAS architecture based on the SIoT paradigm, capable of being implemented on various industrial systems.

3. AN ARCHITECTURE TO IMPLEMENT COLLABORATIVE LEARNING FOR SOCIAL INDUSTRIAL ASSETS

The objective of the MAS architecture proposed here, which consists of three layers, is to implement Collaborative Learning for social industrial assets. The first layer is formed by Virtual Assets, software components that ensure that the data originating from machines is pushed to digital agents in a standardised format, and at regular intervals. The second layer consists of Digital Twins, digital agents that run the algorithms of interest for the asset manager using the standardised data from the Virtual Assets. The third layer of our architecture is the Social Platform, that can be hosted in a central server or in the cloud. All the communications to/ from an agent, and the interactions with the external world, happen via the Social Platform (see figure 1).

3.1 First Layer: assets, Virtual Assets and standardisation of data

In our architecture, the data originating from a physical asset is standardised by a Virtual Asset before being sent to the Digital Twin.

The motivation for the introduction of Virtual Assets is the heterogeneous nature of industrial asset fleets. A manufacturing facility, for instance, may have a milling machine, a packaging machine, and a lathe among many other kinds of machines. They might also come from different manufacturers, serve different purposes, and have different specifications. In another example, an automobile company produces different models of vehicles, which are suitable for different terrains or performances required. The number and types of sensors, or operating conditions, may vary among the vehicles. Virtual Assets are designed to standardise the data coming from these vehicles in a format that is conducive to a generic Digital Twin.

Virtual Assets A Virtual Asset is a software component present for each corresponding physical machine. It is responsible for standardising the asset data before it reaches the Digital Twin. The data from Virtual Assets consists of three main parameters: the Machine Identifier, the Features (with time at which they were recorded), and the Events (kind of event, and time of event). The 'Machine Identifier' gives the asset a specific identity in the asset fleet, and includes information of the asset make, location and operator. 'Features' here refer to sensor generated values. 'Events' can be failures, warnings, user messages, etc. figure 2 describes how the data is made into a standardised format after passing through a Virtual Asset.

Similar enterprise-level solutions exist. For example, MTConnect (Vijayaraghavan et al. (2008)), which standardises the data being transferred across the entire system. Our approach differs from MTConnect, as here the data from each asset is standardised individually by the Virtual Assets and not at fleet-level.

3.2 Second Layer: generic Digital Twins

Extensive instrumentation, increased digitisation and the heterogeneity of manufacturing systems make the design of industrial agents a difficult task. Even in relatively homogeneous asset fleets the make of the assets vary, and so does the nature of the asset's data. Addressing this challenge, the concept of a Virtual Asset presented in section 3.1 enables the development of a generic Digital Twin. These generic Twins are capable of working with standardised data provided from a variety of assets. This is aimed to free the asset manager from the cumbersome task of designing a specific Digital Twin for each of the many kinds of asset present in a typical industrial environment.

A generic Digital Twin The layout of the Digital Twin proposed here is generic, which means, it can be adapted to virtually every type of asset and industrial problem. Figure 1 shows the layout of the proposed Digital Twin. Data flows into the Digital Twin from two sources: its corresponding asset, and the Social Platform; and is stored in a data repository. This data is then used to run a

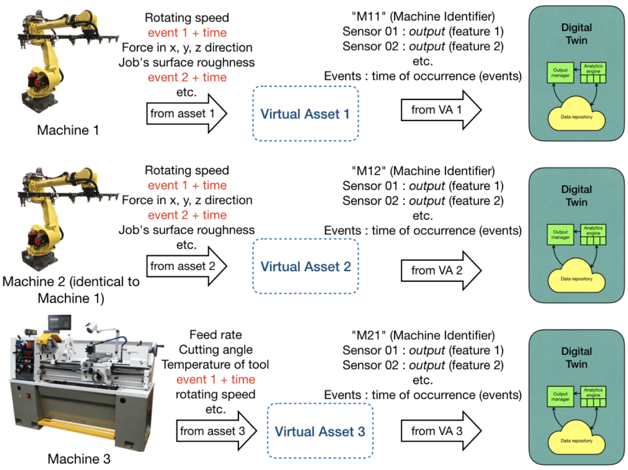

Fig. 2. Virtual Assets are the software components that standardise the data arising from the assets before it is sent to the Digital Twins. Shown in the figure are the three components of standardised data, which enable the Digital Twins to implement the correct algorithms. The figure showcases examples for three different machines of two different kinds, the machine type is specified in the Machine Identifier. In this case the first number in the identifier indicates the machine type and the second the machine ID within this type.

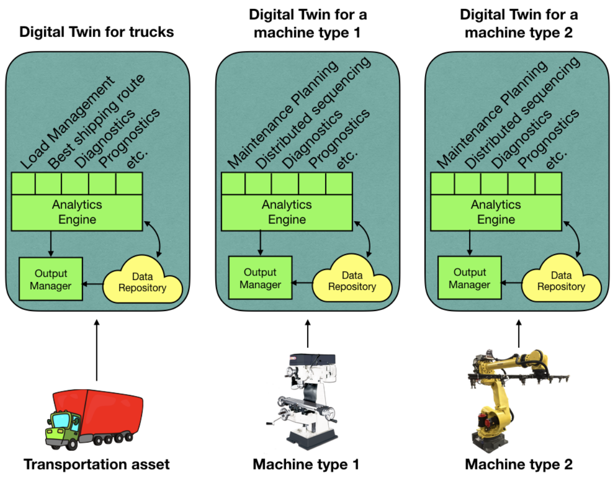

Fig. 3. The analytics engine of a generic Digital Twin model employs different algorithms, depending on the asset it is associated with. The figure describes this for the case of a transportation asset and two other manufacturing machines in an industry.

diverse set of analytic algorithms. These form the Analytic Engine of the Digital Twin. Typically for each *kind* of asset, their associated Digital Twins may run different kind of algorithms. Which algorithms will the Twins run is determined by the needs of the asset manager and conveyed by the Social Platform. An output manager monitors the streams of data flowing out of the twin, and is thus responsible for data sharing and collaboration with its friends in the asset fleet.

The algorithms run by the analytic engine of the Digital Twin may address tasks like health management, performance optimisation, and other optional features which may be particular to that asset type. The figure 3 shows how prognostics and diagnostics are performed for all the assets, but certain tasks like load management and path determination are performed for the transportation assets only. Additionally, algorithms supporting a hypothetical centralised clustering performed by the Social Platform can be implemented in the analytics engine here. For example, a secondary hand-shake distributed clustering algorithm can be implemented on agents to reinforce centralised clustering.

The computing capabilities of the agents allow for a flexible heterarchy of the system. For instance, when a system is in operating state, the algorithms in the Digital Twins keep processing data both from themselves and from collaborating assets. This allows the algorithms (often designed to infer empirically-based models) to learn in real-time. This automatised heterarchy can be stopped at any time by request of the Social Platform.

3.3 Third Layer: the Social Platform

The Social Platform forms the third layer of the proposed architecture. Hosted in a single or multiple servers in the cloud, the Social Platform is both a gateway for human-Digital Twin interaction, and also an enabler for asset to asset communications. The primary capabilities of the Platform are shown in figure 1. These are running enterprise level algorithms which are implemented using data provided by the whole asset network, and storing the relevant system data in a repository. Algorithms implemented on the Platform are focused on performing enterprise level optimisation, and extracting fleet performance trends.

Collaboration among assets becomes efficient when an asset prioritises the data originating from its friends (similar assets). To enable this in our architecture, a matrix comprising of distances (similarities) between assets is formed and stored in the Social Platform. We call it the ‘friendship matrix. As the system operates, inter-asset similarities are calculated at regular intervals, subsequently updating the friendship matrix. Similarity may be calculated based on a variety of indicators such as feature data, machine type, environmental data, etc. Since it is a common channel for the data flowing in the MAS, the Platform theoretically is best informed to calculate similarity metrics. This is done through Enterprise level algorithms such as k-means clustering. Otherwise, decentralised clustering can also be run in the analytics engine of the Digital Twin. The information regarding the asset's clusters will in any case be in stored in the Platform's data repository, in form of a friendship matrix.

For each asset, its cohort of collaborating assets is given by the N closest assets in the friendship matrix. Collaborative learning is then implemented by sharing data between pairs of “friends”. The data received by an asset from a friend may be weighted in the algorithms running in the analytics engine of the Digital Twin according to their estimated similarity.

3.4 Collaborative Learning

In a fleet of assets a rare catastrophic failure may occur only to a small subset of assets. In this case, it would be beneficial to convey the information regarding this failure to the other similar assets in the fleet. If this is not done, we might face a scenario where an event, although already known to the fleet, would be unknown to machines which have not encountered it yet. Thus their algorithms will fail to predict it. This example becomes especially relevant for new machines being added to a fleet of old ones. In a collaborative MAS architecture, the trajectories corresponding to a newly registered event can be shared among similar assets and other agents can thus be made aware of such circumstances in future.

To make this inter-asset collaboration efficient, it is crucial to ensure that the data being shared covers all relevant information, and at the same time, is not bulky. To achieve this, the assets keep sharing certain pre-defined performance parameters at regular intervals while the asset is in normal condition. As soon as an asset encounters a certain *new event of interest*, the data corresponding to that time frame, i.e. a trajectory to that event, is shared as a 'new training data-set' for other assets. Subsequently, specific causes and analysis of the event are shared among Digital Twins too.

Apart from making the system more robust, collaborative learning makes a system agile and more efficient. For instance, through collaboration, machines of a manufacturing unit can actively manage their load, by continuously evaluating their health condition and comparing with that of other similar machines. A healthy machine is capable of producing more output, thus reducing the production load of deteriorated machines and the maintenance downtime.

4. AN ILLUSTRATIVE EXAMPLE: CLUSTERING AND PROGNOSTICS IN THE C-MAPPS DATA-SET

We demonstrate the use of the above described architecture to determine the Remaining Useful Life (RUL) for a fleet of turbofan engines. Due to its link to Condition Monitoring, calculating an asset's RUL is a problem that combines asset management and IIoT technologies. Here, we use the C-MAPSS (Saxena and Goebel (2008)) dataset to showcase collaborative RUL estimation. The dataset consists of four fleets of engines, which are labelled as FD001-4. For our example we'll be using fleets FD001 and FD003 only. Engines within FD001 and FD003 share the same operating conditions, with the difference being that engines in FD003 fail due to High Pressure Compressor degradation and fan degradation while engines in FD001 only present the first kind of failure. The data-set employed here consist of multi variate time-series in the form of rows of sensor data recorded after fixed time-steps. Each machine starts normally, develops fault during operation, this fault grows in magnitude and the machine eventually fails. Both FD001 and FD003 feature 100 independent trajectories to failure. We group 20 trajectories to failure together, in each fleet, to simulate multiple machines. This is not ideal, but since all the machines in a fleet are identical, it is sufficient to serve as an example. Collaboration is implemented simply by sharing failure trajectories among clusters of machine 'friends'.

In our example, the architecture is implemented using the python programming environment. For our case, Virtual Machines directly read the data of their corresponding engines from csv sheets, instead of receiving data from a real asset. To simulate real-time operation, the VMs read the data at fix time intervals, and are unaware of what lies ahead. The data, after being pushed to the Digital Twins using the 'socket' library, is processed by a naive prognostics algorithm, based on a fixed window K-Neighbors classifier from the 'sklearn' machine learning library. In short, the data coming from the asset's sensors is classified according to its known remaining useful life based on the width of a predefined time window (see Table 1). This classification algorithm is then used to make approximate predictions about the RUL of new trajectories.

Table 1. Example of a fixed-window classification, after normalisation of the sensor values, a class is assigned to each data point with regards to its remaining useful life. In green, classes assigned *a posteriori* of the known failure (marked in red). In blue, classes predicted by the classification algorithm.

Sensor1	0.9	0.7	0.55	0.99	1	0.3	0.9	0.71
Sensor2	0.87	0.77	0.99	1	1	0.2	0.88	0.77
Sensor3	1	0.2	1	0	1	0.1	1	0
RUL	5	4	3	2	1	0	?	?
Class	2	2	1	1	0	0	2	2

Apart from the prognostics algorithm implemented in the Digital Twins, a basic K-Means clustering algorithm is implemented on the Social Platform, aimed to determine the friendship matrix. This algorithm, identifies and clusters similar engines based on Euclidean distances between the sensor values. This has been implemented to illustrate the role of the Social Platform and the Digital Twin's output manager. The output managers share the average of the data points received from their corresponding VMs in a time-step, with the Social Platform, which serves as a statistical indicator of the status of the asset. Then, the Social Platform uses these points to determine the Friendship Matrix of the assets using the centroid based clustering approach (figure 4). Once the clusters are stable, in each time step, the Social Platform then shares every asset's data with its friends, enabling every asset with large amount of data from its cluster.

After implementing the example described above and quantifying the results obtained for collaborative and non-collaborative approaches, we find the significantly higher accuracy of Collaborative approach as evidence for the efficiency of our architecture and proof of the advantages of collaborative learning (figure 5).

5. CONCLUSION

We propose a Multi Agent System architecture specially designed to implement collaborative learning in social industrial assets. The proposed architecture is envisaged within the paradigm of the Social Internet of Industrial Things (SIoT), and allows assets within an industrial system to self-assemble in networks of collaborating assets. The architecture allows for any kind of collaboration stemming from data sharing among similar machines. We implement the proposed architecture for collaborative prognostics, using a naïve fixed-window prognostics algorithm

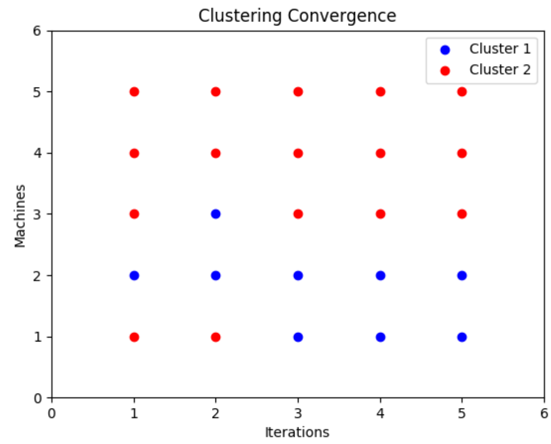

Fig. 4. The Social Platform is responsible for forming clusters of similar assets. In our example, we form two machines from FD001 (cluster 1) and three from FD003 (cluster 2). This plot displays the convergence and formation of stable clusters after 5 iterations, with the machines being grouped correctly based on the example.

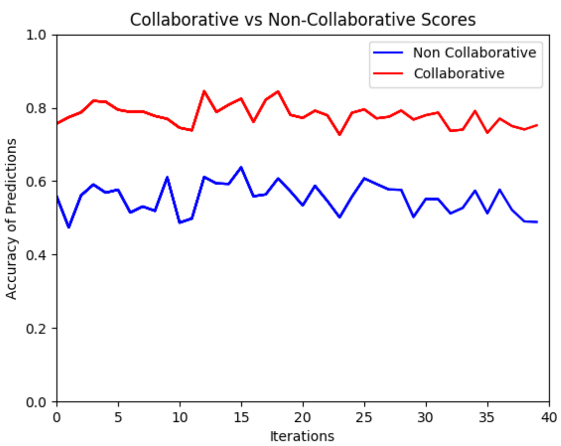

Fig. 5. Accuracy of the fixed-window predictive algorithm for a machine of Cluster 1 in the Collaborative (red) and Non-collaborative (blue) cases.

to predict RULs in the C-MAPPS data-set. We show how assets cluster in stable groups of friends that then exchange data to improve the prognostics algorithm. The distributed nature of the Digital Twins is replicated by using different computers to represent different assets. Initial results show that continuous exchange of data between similar assets significantly improves prediction accuracy. Future work will consist on a real-case implementation with multiple failure modes, asset kinds and analytic engines in the Digital Twins.

REFERENCES

Alonso, E., D'inverno, M., Kudenko, D., Luck, M., and Noble, J. (2001). Learning in Multi-agent Systems. *The Knowledge Engineering Review*, 16(3), 277–284.

Atzori, L., Iera, A., and Morabito, G. (2014). From "smart objects" to "social objects": The next evolutionary step of the internet of things. *IEEE Communications Magazine*, 52(1), 97–105.

Bagheri, B., Yang, S., Kao, H.A., and Lee, J. (2015). Cyber-physical systems architecture for self-aware machines in industry 4.0 environment. In *IFAC-PapersOnLine*, 1622–1627.

Christensen, J.H. (2003). HMS/FB Architecture and its Implementation. In *Agent Based Manufacturing: Advances in the Holonic Approach*. Springer-Verlag Berlin Heidelberg.

Evans, P.C. and Annunziata, M. (2012). Industrial Internet: Pushing the Boundaries of Minds and Machines. *General Electric*.

Ganchev, I., Ji, Z., and O'Droma, M. (2016). Designing a low-cost data transfer unit for use in IoT applications. In *2016 8th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, 85–88.

GE Digital (2017). The Digital Twin: Compressing Time to Value for Digital Industrial Companies. Technical report, General Electric.

Giordani, S., Lujak, M., and Martinelli, F. (2013). A distributed multi-agent production planning and scheduling framework for mobile robots. *Computers & Industrial Engineering*.

Giret, A. and Botti, V. (2004). Holons and agents.

Jeschke, S., Brecher, C., Meisen, T., Özdemir, D., and Eschert, T. (2017). *Industrial Internet of Things and Cyber Manufacturing Systems*, 3–19. Springer International Publishing, Cham.

Lee, E.A. (2008). Cyber Physical Systems: Design Challenges. In *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*, 363–369.

Li, H., Salvador Palau, A., and Kumar Parlikad, A. (2018). A Social Network of Collaborating Industrial Assets. *Submitted to: Journal of Risk and Reliability.*

Li, S., Xu, L.D., and Zhao, S. (2015). The internet of things: a survey. *Information Systems Frontiers*, 17(2), 243–259.

Mönch, L. and Drießel, R. (2005). A distributed shifting bottleneck heuristic for complex job shops. *Computers and Industrial Engineering*.

Ning, H. and Wang, Z. (2011). Future internet of things architecture: Like mankind neural system or social organization framework? *IEEE Communications Letters*, 15(4), 461–463.

Saxena, A. and Goebel, K. (2008). Turbofan Engine Degradation Simulation Data Set. *NASA Ames Prognostics Data Repository (http://ti.arc.nasa.gov/project/prognostic-data-repository), NASA Ames Recearsh Center, Moffett Field, CA*, NASA Ames.

Unland, R. (2015). Industrial Agents. In *Industrial Agents*, 23–44. Elsevier.

Vijayaraghavan, A., Sobel, W., Fox, A., Dornfeld, D., and Warndorf, P. (2008). Improving Machine Tool Interoperability Using Standardized Interface Protocols : MT Connect. In *2008 International Symposium and Flexible Automation,*, 21–24.

Xu, L.D., He, W., and Li, S. (2014). Internet of things in industries: A survey.